

BITS & P.C.s
COMPUTER PRODUCTS

18 RYE GARTH WETHERBY
WEST YORKSHIRE
LS22 4UL

TEL 0937 63744

```
**** * *** ****      *** **   ****
* * * * *      *   * * * *
**** * *      **** ***  *** *   ****
* * * *      * *   *   *   *   *
**** * *      ****      * *   ** * ****
```

TOOLKIT

FACILITIES	PAGE No
APPEND	2
AUTO LINE NUMBER	3
DELETE	4
DUMP	5
FIND	6
HELP	7
HEX	8
INKEY	9
RINK	9
KEYBOARD REPEAT	9
MON	10
PRINTER HANDSHAKE	10
RENUMBER	11
STEP	12
INSTALLING YCUR TOOLKIT(eprom)	13
LOADING YOUR TOOLKIT (tape)	14
SCRATCHPAD USAGE	15

AUTO LINE NUMBER

FORMAT: AUTOmm,nn

PURPOSE: To generate a line number automatically after every entry.

REMARKS: AUTO begins numbering at (mm) and increments each subsequent line number by (nn). The default for both values is 10. To escape from AUTO type escape (shifted enter).....

EXAMPLE: AUTO100,10 generates line numbers 100 110 etc.

The use of the AUTO command to produce evenly spaced line numbers does have a useful end product. That is spare line numbers are left between each line used. This allows for the inclusion of extra commands at some later time as your program develops. After adding these extra lines it is always possible to RENumber the new code so as to restore the tidy appearance. One further use of the AUTO command is shown below.

EXAMPLE.

```
1000 REM PRINT SUBROUTINE
1010 REM
1020 REM END PRINT SUB.
```

Upon hitting enter in line 1020 the auto number command would produce 1030. If you wish some other line number to follow ie 2000 then simply backspace over 1030 then type 2000 and enter your line, the auto number routine will then continue at 2000 + inc.

DELETE

FORMAT: DELEmm,nn

PURPOSE: To delete program lines.

REMARKS: This command will erase all lines between the limits stated but will not error if either of the limits stated does not exist, the command will simply delete all valid lines included. It should be remembered that DELE will not remove references such as GOTO nn, IF THEN nn . where nn is a line number that has been deleted.....

EXAMPLE: DELE10,50 deletes lines 10 to 50 inclusive
 DELE,100 deletes all lines up to 100

By using both the on screen editor and the DELE command a section of program can be moved relative to the main program.

10 nnn nnn nnn nnn	this section of code may need to
20 nnn nnn nnn nnn	be moved say to 1000 to 1060.
30 nnn nnn nnn nnn	
40 nnn nnn nnn nnn	By using the on screen editor
50 nnn nnn nnn nnn	this section of code can be
60 nnn nnn nnn nnn	renumbered into the range

required. Then the DELE command can be used to delete the original section of code.

DELE10,60

DUMP

FORMAT: DUMP

PURPOSE: To list on screen all numeric and string variables defined in a program. With the exclusion of arrays.

REMARKS: The DUMP command makes use of a pointer provided by the BASIC, this pointer to the table of variables is only accurate if the program text is unaltered after the program has been RUN. This command will display up to twenty one lines of variables at any one time. Press space to display further information. Press "escape" to leave the command.

EXAMPLE: DUMP

```
A = 1:    B = 2:  C = 3:
CD = 3.14159: Z = 100 :
D = 1234: T = 1010 :
AA$="DUMP"
```

FIND

FORMAT: FIND

PURPOSE: To find the first appearance of a string in the program.

REMARKS: After typing FIND and entering, the command will prompt for an input string, you may then enter the string you wish to find (up to a maximum of 48 characters). After entering, the computer will find the first occurrence of the string entered above. ESCAPE will find the next occurrences. It is possible that the first appearance of your string is one of the reserved words in the BASIC program, to prevent the command finding reserved words enter CONTROL B (see example below). One further point if the string you enter is "FOR" the command can also find "FORM" or "FORMAT".

OFF:

To disable the above command type OFF. The computer will now return to the command level.

HELP

FORMAT: HELP(nn)

PURPOSE: To assist in the isolation of errors after the BASIC has stopped, giving an error message .

EXAMPLE:

HELP :The offending line will be displayed on the screen, the cursor will point to the statement causing the problem.

HELPnn :This mode of operation allows you to display a line specified by (nn). This when used with CONTROL B will enable you to display a line in compressed form, it is then a simple matter to add to the line thus making it easier to enter lines more than 48 characters long.

HEX

FORMAT: HEXnnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn nnnn

PURPOSE: This command is used to convert up to 10 HEX numbers to their decimal values.

REMARKS: After conversion the decimal values can easily be "edited" into a data statement, to help in this each conversion result is separated by a comma. Thus by simply "editing" the line number and DATA into the above there is a ready made DATA statement.

EXAMPLE:

HEX0C80 0D00(enter)

3200,3328

10 DATA 3200,3328

INKEY & RINK

These are not extra commands as such but they do provide a very much simplified method of giving these facilities.

INKEY

This routine scans the keyboard until a key is pressed, when a key is pressed this routine will return the ASCII value of that key.

To activate the INKEY routine the line below should be entered.

```
10 DOKE 4100,DEEK(3201)+1959
```

RINK

This routine will scan the keyboard once and then return. If a key has been pressed then the ASCII value of that key will be returned, if no key has been pressed then 0 will be returned.

To activate the RINK routine the line shown below should be entered

```
10 DOKE 4100,DEEK(3201)+1949
```

KEYBOARD REPEAT

The repeat facility is automatically activated upon entry into the TOOLKIT. The repeat waits for one second and then begins repeating at ten per second.

K8 is used to disable the repeat if required

K0 can then be used to re-establish this facility later

These values can be called from BASIC if you have passed into the BASIC via the TOOLKIT. This is done by poking the value in the normal way i.e. 10 POKE 3111,8 or 10 POKE 3111,0.

REMARKS: Control of the keyboard facilities has been given to the TOOLKIT. This change means that K commands should only be used when under TOOLKITS control.

FORMAT: MON

PURPOSE: To exit from BASIC to NAS-SYS

REMARKS: Typing MON or MONITOR whilst in BASIC will cause a return to NAS-SYS.

NOTE: The new short form for return to NAS-SYS should now be used at all times. The short method shown in the NASCOM 2 manual ie (control,graph,left square bracket) will not now function correctly.

PRINTER HANDSHAKE

The handshake routine is automatically enabled on entry to the TOOLKIT. All that then remains to complete the process is to activate the normal printer routine i.e. X0

This program sets up the UOUT jump, so that whenever output to the printer is required handshake is enabled.

This program assumes that the BUSY line from the printer is connected to bit 7 of the Keyboard input port (PORT 0).

If at some time you wish to disable the printer handshake enter X40. To re-enable afterwards enter X0.

NOTE. A small hardware modification to the NASCOM 2 main board is required to complete the handshake. Reference should be made to the manual supplied with your IMP printer.

REMARK: As the Output routine is now being controlled by the TOOLKIT, no attempt should be made to modify the output routine when in NAS-SYS.

RENUMBER

FORMAT: RENU(new number),(increment)

PURPOSE: To renumber program lines , and references to them.

REMARKS: (new number) is the first line number to be used in the new sequence. The default value is 10. (increment) is the increment to be used in the new sequence .The default value is 10.

RENU also changes all line number references following GOTO , GOSUB , THEN etc. If a nonexistant line number appears in a statement the renumber command will replace that line number with 65529 .This "odd" number being the highest line number that BASIC will allow. Renu cannot be used to change the order of program lines. RENUmber will also trap an attempt to renumber with values that would cause an overflow ie a line number greater than 65529. For example RENU 1000,1000 on a 250 line program would result in an overflow message

EXAMPLES: RENU

Renumbers the whole program
The first new line number
will be 10 ,the lines incremented
by 10.

RENU100,10

Renumber the whole program
The first new line number will
be 100. the lines incremented by 10

STEP

FORMAT STEP

PURPOSE: STEP can be used to monitor the value of one or more of the variables in use.

REMARKS: After entering the STEP command the computer will prompt for an input string. This string could simply be A,B,C. The value of these three variables would then be displayed on screen. A second approach would be to enter a string of the type shown in the second example below.

EXAMPLE 1

```
STEP(enter)
STRING?
X(enter)
RUN(enter)
```

The above will now step through the BASIC program one statement at a time. In the top left of the screen the line number just executed will be displayed followed by the value of the variable X.

EXAMPLE 2

```
STEP(enter)
STRING?
"The value of X=X(enter)
RUN(enter)
```

In this second example the string entered will now be displayed between the line number and the variable. The maximum number of characters allowed in the string is 48.

OFF To break from the step mode, break from the program in the normal way then type OFF

NOTE The use of the repeat keyboard by holding a key down permanently will provide a method of rapidly stepping through a program.